A. Additional Details on Environments

We provide a detailed introduction to the experimental environments used in this study.

A.1. Points24

State and action space. At each state s_t in the Points24 task, the agent observes an image showing four poker cards and a text-based representation of the current formula. The goal is to form a formula equal to 24 using the numbers represented by the four cards and basic operators. Card "J", "Q", "K" are all treated as number 10. The action space includes {"1", "2", ..., "10", "+", "-", "*", "/", "(", ")", "="}, and each card can only be used once. Selecting a number not present in the image or one that has already been used is considered an illegal action. If the action is legal, the corresponding number or operator is appended to the current formula, forming the next state s_{t+1}; if the action is illegal, the state remains unchanged $s_{t+1} = s_t$. The environment does not guarantee that the four cards in the image have a feasible solution equal to 24.

Reward function. At each step, the agent receives a reward $r = -1$ for outputting an illegal action and a reward $r = 0$ for a legal action. The episode terminates when the agent outputs "=" as an action or the step counts exceeds $T = 20$. At termination, if the formula evaluates to 24, the agent receives an outcome reward $r = 10$; otherwise, it receives $r = -1$.

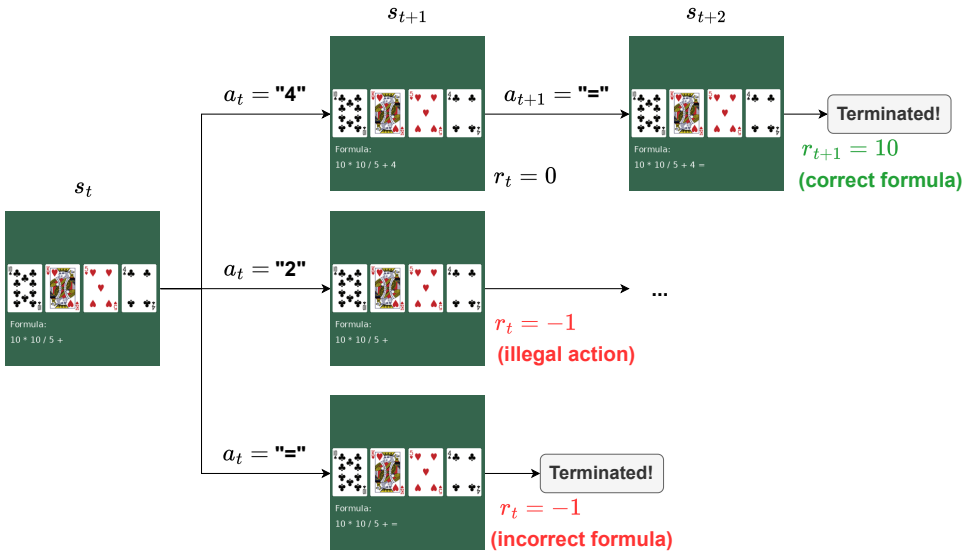

Figure 1. **The Points24 task.**

A.2. ALFWorld

State and action space. In the ALFWorld environment in our experiments, the agent receives an RGB observation image and a history of past actions at each state s_t. The action space includes all possible interactions in the current scenario, typically categorized as: (1) `go to {recep}`, (2) `take {obj} from {recep}`, (3) `put {obj} in/on {recep}`, (4) `open {recep}`, (5) `close {recep}`, (6) `toggle {obj} {recep}`, (7) `clean {obj} with {recep}`, (2) `heat {obj} with {recep}`, (2) `cool {obj} with {recep}`, where `{obj}` and `{recep}` denote objects and receptacles. After an admissible action is taken, ALFWorld renders the updated scene from the agent's view as the next state s_{t+1}. $s_{t+1} = s_t$ if the action is illegal.

Notably, the original ALFWorld environment provides a text description of the scene in each state without the action history. However, to prevent the agent from relying on the textual description rather than visual observation and to better simulate real-world scenarios, we modified the state by removing the text description and adding the action sequence taken. This adjustment increases the difficulty, emphasizing the agent's visual recognition and long-horizon decision-making capabilities.

Reward function. The reward system of ALFWorld consists of two components. Each state s has a set of admissible actions $\mathcal{A}_{\text{adm}}(s)$, and illegal actions are penalized. Additionally, each task in ALFWorld has both the final goal g_{task} and sub-goals g_{sub},

and achieving these goals also provides rewards. Formally, the reward function can be written as:

$$r(s_t, a_t, s_{t+1}|g_{\text{task}}) = 50 \times \mathbf{1}(s_{t+1} = g_{\text{task}}) + \mathbf{1}(s_{t+1} = g_{\text{sub}}) - \mathbf{1}(a_t \notin \mathcal{A}_{\text{adm}}(s)). \quad (1)$$

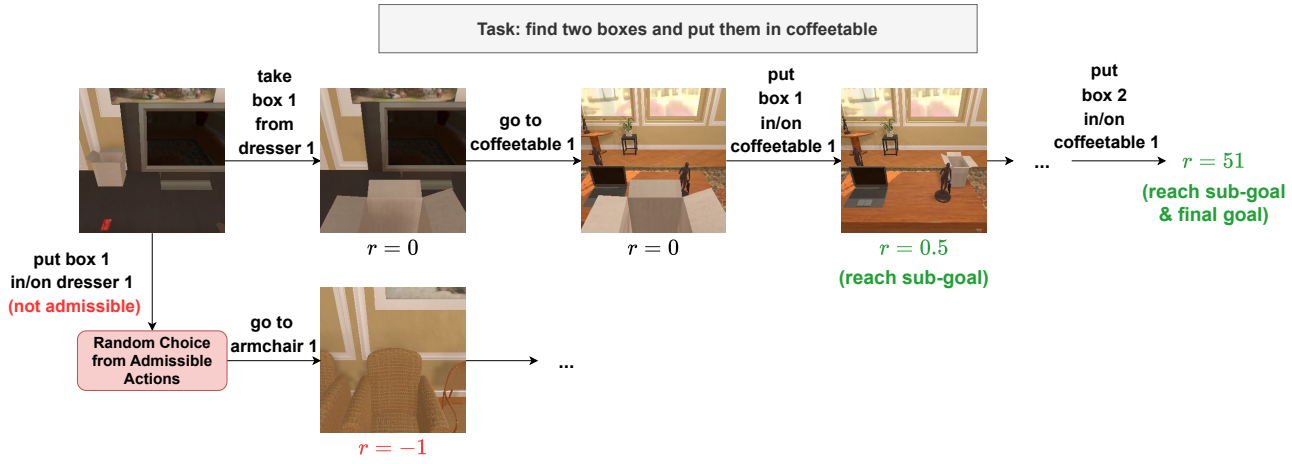

Figure 2. **The ALFWorld task.**

A.3. Other Games in the *gym_cards* Environment

We briefly introduce the other tasks in the `gym_cards` environment, which have significantly smaller state and action spaces, shorter episode lengths, and lower complexity than the two tasks we selected. As a result, these tasks do not exhibit the thought collapse phenomenon. Nevertheless, GTR still achieves performance improvements in these more straightforward tasks.

Numberline. The agent receives an image displaying the text "Target: x" and "Current: y", where x, y are integers in $[0, 5]$. The action space is {+, -}, which increments or decrements the current number by 1, respectively. The goal is to make the current number equal to the target number. The agent gets a reward of 1 upon achieving the goal and a penalty of -1 if an action moves the current number away from the target. The game can always be solved within 5 steps.

EZPoints. This task is a simplified variant of Points24, with the image containing only two cards, the available operators limited to {+, -, =}, and the target value is 12. In addition, the EZPoints environment guarantees that the two cards in the image always have a valid solution. The correct formula always takes 4 steps.

Blackjack. The task is to win the blackjack game. The image at each state includes two cards of the dealer (one of them facing down) and all cards from the player. The action space is {"stand", "hit"}. The agent gets one more card when choosing "hit", and the game terminates when choosing "stand". Theoretically, the player has an expected winning rate slightly below 50%.

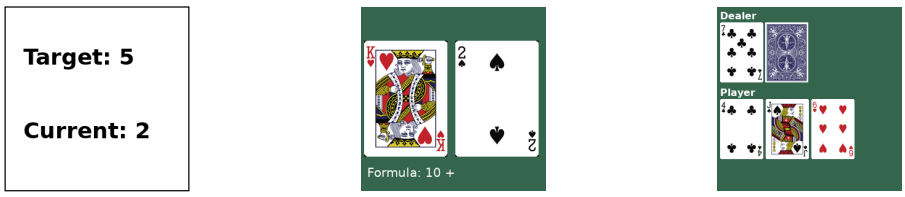

Figure 3. **Other tasks in the `gym_cards` environment.**

B. Additional Details on Training

B.1. Training Setting

Drawing inspiration from the RLHF training framework [5] and prior related work [7], we perform one epoch of supervised fine-tuning on the base LLaVA-v1.6-mistral-7B model [2–4] before RL training, which is referred to as *LLaVA-sft* in the results. The datasets are sourced from the RL4VLM paper [7], with labels for the `gym_cards` environment provided by a task solver and labels for the ALFWorld environment generated by GPT-4V.

B.2. Hyperparameters

In Table 1, we provide the hyperparameter settings used for GTR training, which are primarily derived from values proposed in previous work [7]. We employ LoRA [1] to fine-tune the entire VLM model, including the CLIP vision encoder [6], LLM backbone, and MLP projector.

Hyperparameter	Value
General Setup - Training	
Learning rate	`CosineAnnealingLR`
Initial learning rate	$1e-5$
Final learning rate	$1e-9$
Maximum learning rate step	25
Discount factor γ	0.9
GAE λ	0.95
PPO entropy coefficient	0.01
PPO value loss coefficient	0.5
PPO clip parameter c	0.1
PPO epoch	4
Gradient accumulation steps	128
LoRA r	128
LoRA α	256
LoRA `dropout`	0.05
General Setup - Models	
Generation max text length	256
Generation temperature	0.2
Generation repetition penalty	1.2
Corrector max text length	600
Corrector temperature	0.4
For Points24 task	
Environmental steps	15000
Thought probability coefficient	0.5
For ALFWorld task	
Environmental steps	5000
Thought probability coefficient	0.2

Table 1. **Hyperparameters of GTR**

B.3. Computational Overhead

Running a large corrector model at each RL step introduces additional computational overhead during training. To provide a more comprehensive comparison of corrector models of different types and sizes, we evaluate the performance, cost, and training time of both open-source and closed-source models, as well as large and small models. The results in Table 2 show that although open-source models can reduce overhead, the performance of Qwen2.5-VL-72B is undermined by its sub-optimal tool-use capabilities, and the 7B version fails to follow proper correction formats.

Corrector Model	Performance	Token Usage (Cost)	Time
GPT-4o	17.5%	33.5M (˜$463.5)	86h
Qwen2.5-VL-72B	6.5%	33.8M (˜$91.6)	110h
Qwen2.5-VL-7B	N/A	31.2M (˜$19.9)	56h

Table 2. **Ablation study on computational overhead across different corrector models.**

C. Additional results of ALFWorld with textual observation

We observe that RL4VLM's performance on ALFWorld in its original paper is attributed to precise textual observations, which notably compensates for the agent's limited visual recognition and reasoning capabilities. This is also the primary reason why we remove the text description in our experiments. Nevertheless, we include the performance of RL4VLM and GTR with the presence of textual observations. The results in Table 3 demonstrate that GTR does not need textual observations but achieves competitive performance as the corrector effectively bridges the gap between vision and text.

	Success Rate
RL4VLM w/ text	21.7%
RL4VLM w/o text	5.4%
GTR w/ text	21.0%
GTR w/o text	17.8%

Table 3. **Performance comparison of ALFWorld with and without textual observations.**

D. Continual Training of GTR

To evaluate the performance of the GTR algorithm over extended training durations, we present results of GTR trained for 30k steps on the Points24 task. As shown in Figure 4, GTR is able to maintain excellent performance.

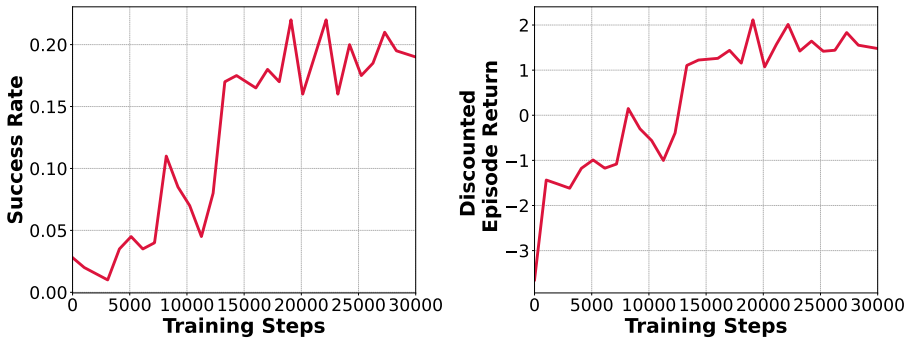

Figure 4. **Training the VLM agent with GTR for 30,000 steps.**

E. Prompts of the Corrector Model

Prompt adopted by the VLM corrector model for the Points24 task

System Prompt: You are an expert 24-point card game player. You are observing four cards in the image and the current formula. The goal is to output a formula that evaluates to 24, and each number can only be used once. The number or operator include ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '+', '-', '*', '/', '(', ')', '='], and the chosen number or operator will be appended to the current formula to reach the correct target formula that evaluates to 24. Note that 'J', 'Q', and 'K' count as '10'.
Query: You will be given the current formula, the thought of a player playing this game, and a target formula. The player's thought may be wrong, please evaluate its correctness by the following aspects:
(1) What are the four cards in the image? If the target formula is 'NOT DETERMINED', use the 'find_all_correct_formulas' tool function to find all possible correct formulas by the four cards in the image. Remember the correct formulas, and do not output the result.
(2) What are the recognized card ranks in the thought? According to the rules, does the ranks in the thought match your observation in question (1), regardless of the order?
(3) What is the proposed formula the player is trying to reach in the thought? Does the proposed formula match the target formula or, if the target formula is 'NOT DETERMINED', one of the possible correct formulas in question (1)?
(4) Does the player choose the correct action to reach the proposed formula or choose '=' if the current formula is complete?
Please briefly answer the above questions, then give your final evaluation. If the thought is incorrect, use all available information for thought correction: determine the next single number or character to append to the current formula and finally provide the correct thought.
Your response should be a valid json file in the following format: {
"answer1": {Text, answer to the first question},
"answer2": {Text, answer to the second question},
"answer3": {Text, answer to the third question},
"answer4": {Text, answer to the third question},
"evaluation": {YES or NO},
"possible_solution": {YES or NO, indicating whether there is a possible solution. None if the thought is correct},
"target_formula": {The given target formula if it is not None. The proposed formula in the thought if the thought is correct. Otherwise, choose an appropriate target formula from all possible correct formulas obtained from the tool function for the player to reach. },
"correction": {Json object, the correct thought. None if the thought is correct}
}
[Current Formula] ...
[Thought] ...
[Target Formula] ...

Prompt adopted by the VLM corrector model for the ALFWorld task

System Prompt: You are an expert in the ALFRED Embodied Environment. The environment requires the player to navigate, take certain objects, interact with objects if necessary, and finally put objects in the designated place to complete the task.
Query: You will be given the visual observation and thought of a player in this environment. The task is to ... You are also given the previous actions the player has taken: ... All admissible actions of the current situation are: ...
Please evaluate if the reasoning is correct by the following aspects:
(1) What objects are in your sight and whether you are holding a certain object? Does the thought correctly identify the image?
(2) Based on the task description and the action history, what should be the player's next sub-goal (notice that the tasks require the player to first pick up certain objects, interact with receptacles if the task is cooling, heating, cleaning or looking in light, and finally placing the object)? Does the thought align with the sub-goal?
(3) Based on the task description and the action history, does the player choose one of the admissible actions to reach the sub-goal? Does the action take effect? If the target object is not in sight, go to an unexplored location; if there is a required object, take it; if the task requires cooling, heating, cleaning, or looking in light, navigate and interact with the receptacles.
Please briefly answer the above questions, then give your final evaluation. If the thought is incorrect, include all available information for thought correction: choose one correct step from the admissible actions for the player to finish the task, and finally provide the correct thought.
Your response should be a valid json file in the following format: {
"answer1": {Text, answer to the first question},
"answer2": {Text, answer to the second question},
"answer3": {Text, answer to the third question},
"evaluation": {YES or NO},
"correction": {Json object, the correct thought. None if the thought is correct}
}
[The player's thought] ...

F. Example Trajectories of Thought Collapse and GTR

In this section, we present real episode examples from the Points24 and ALFWorld tasks that exhibit thought collapse during RL training. These examples illustrate how the model's outputs deteriorate, becoming state-irrelevant and irrational, thereby significantly impairing the agent's decision-making capabilities. This catastrophic phenomenon is the core issue addressed in our work. We also show the GTR-generated trajectories that effectively solve the problem.

F.1. Points24

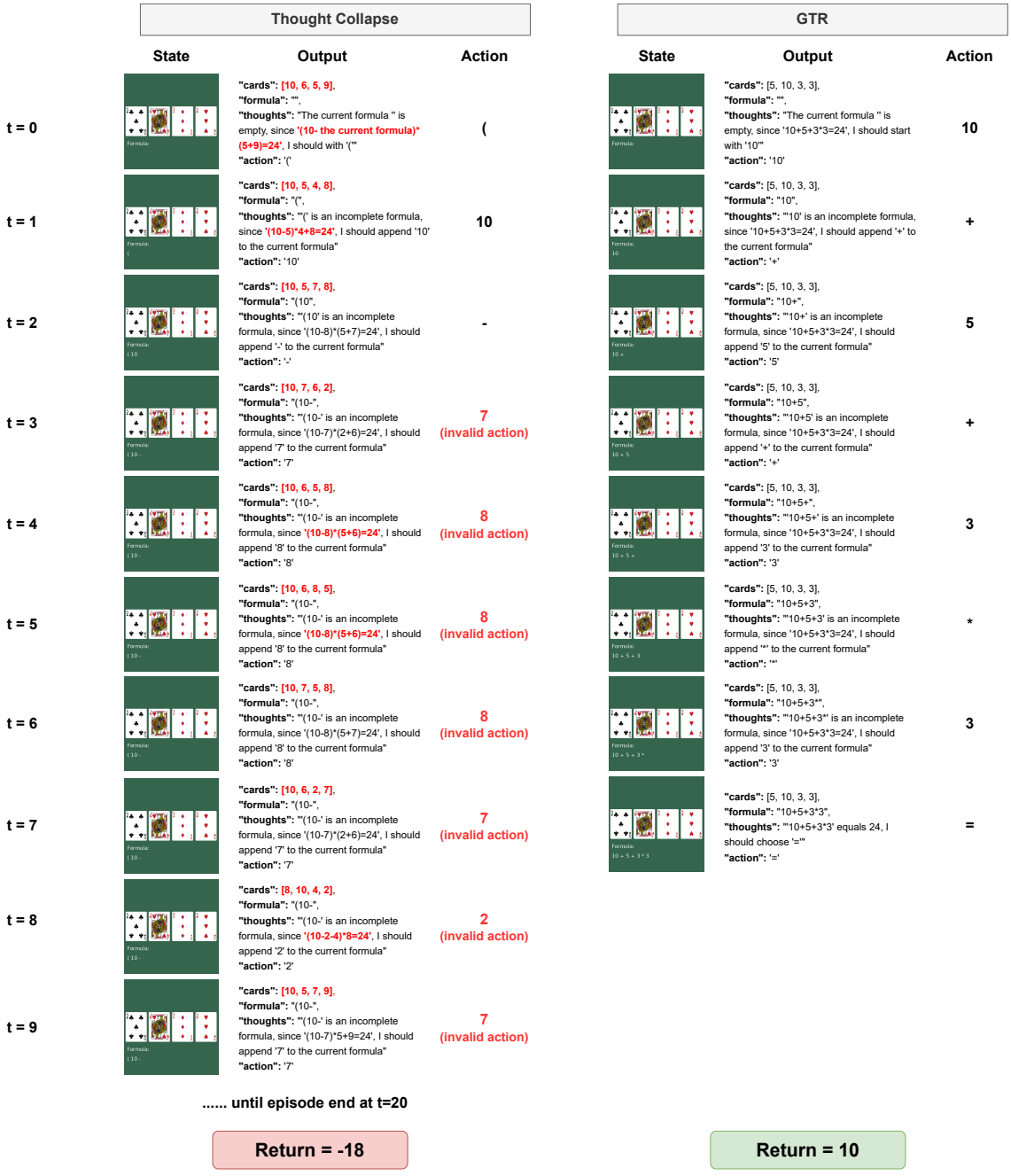

Figure 5. **Examples of agent trajectory produced by RL4VLM (left) and GTR (right) in Points24.**

F.2. ALFWorld

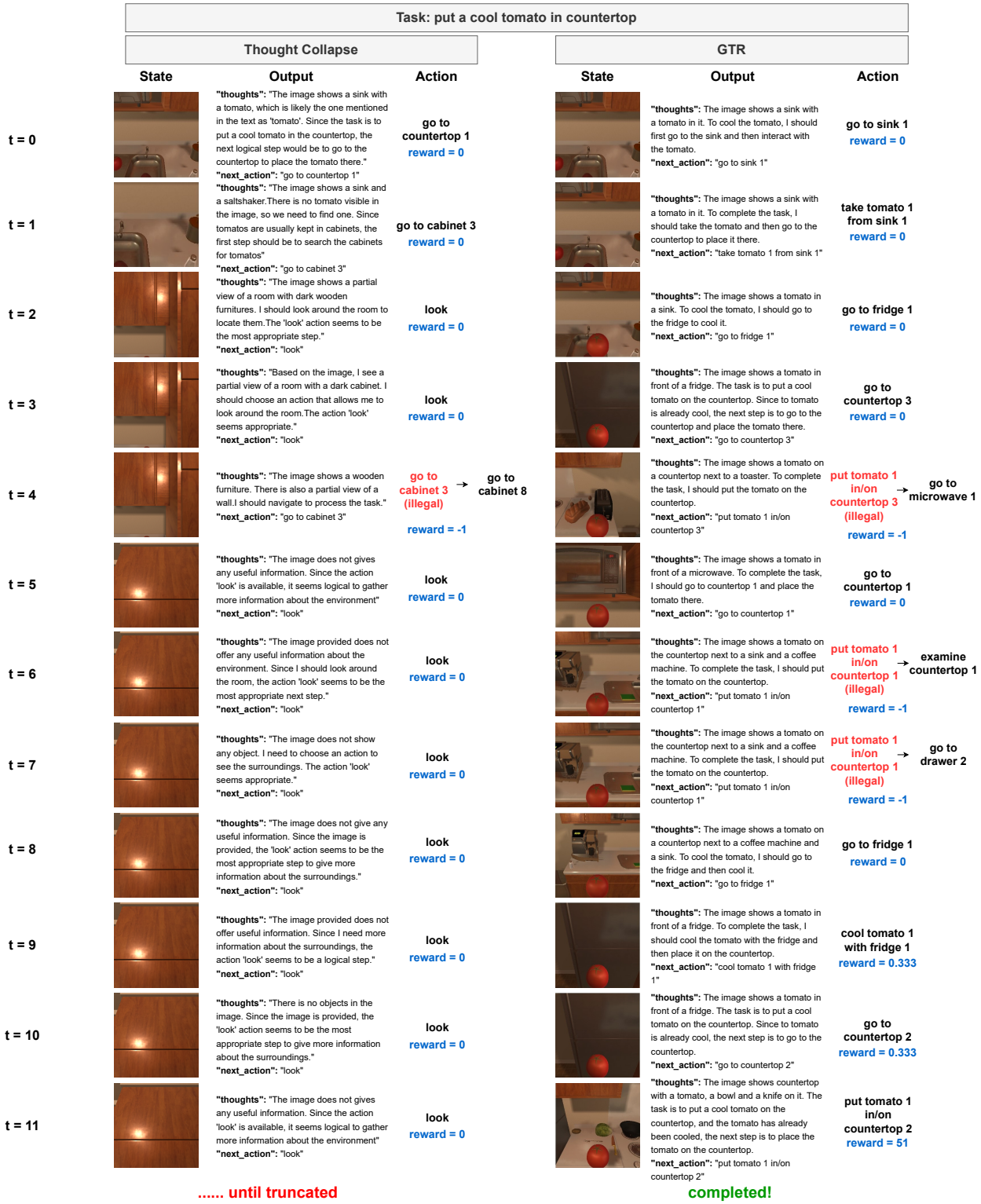

Figure 6. **Examples of agent trajectory produced by RL4VLM (left) and GTR (right) in ALFWorld.**

References

[1] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[2] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 3

[3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[4] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 3

[5] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 3

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 3

[7] Simon Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Peter Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Advances in neural information processing systems*, 37:110935–110971, 2025. 3